Microsoft DevOps Solutions: Developing Security and Compliance

UNDERSTANDING SECURITY AND COMPLIANCE SCANNING

Neil Morrissey
SOLUTIONS ARCHITECT

@morrisseycode   www.neilmorrissey.net

Exam Objectives

Automate dependencies scanning for security (container scanning, OWASP)

Automate dependencies scanning for compliance (licenses: MIT, GPL)

Assess and report risks

Design a source code compliance solution (e.g. GitHub Code scanning, GitHub Secret scanning, pipeline-based scans, Git hooks, SonarQube, Dependabot, etc.))

Visual Studio Marketplace

Agent

Azure DevOps

Pipelines

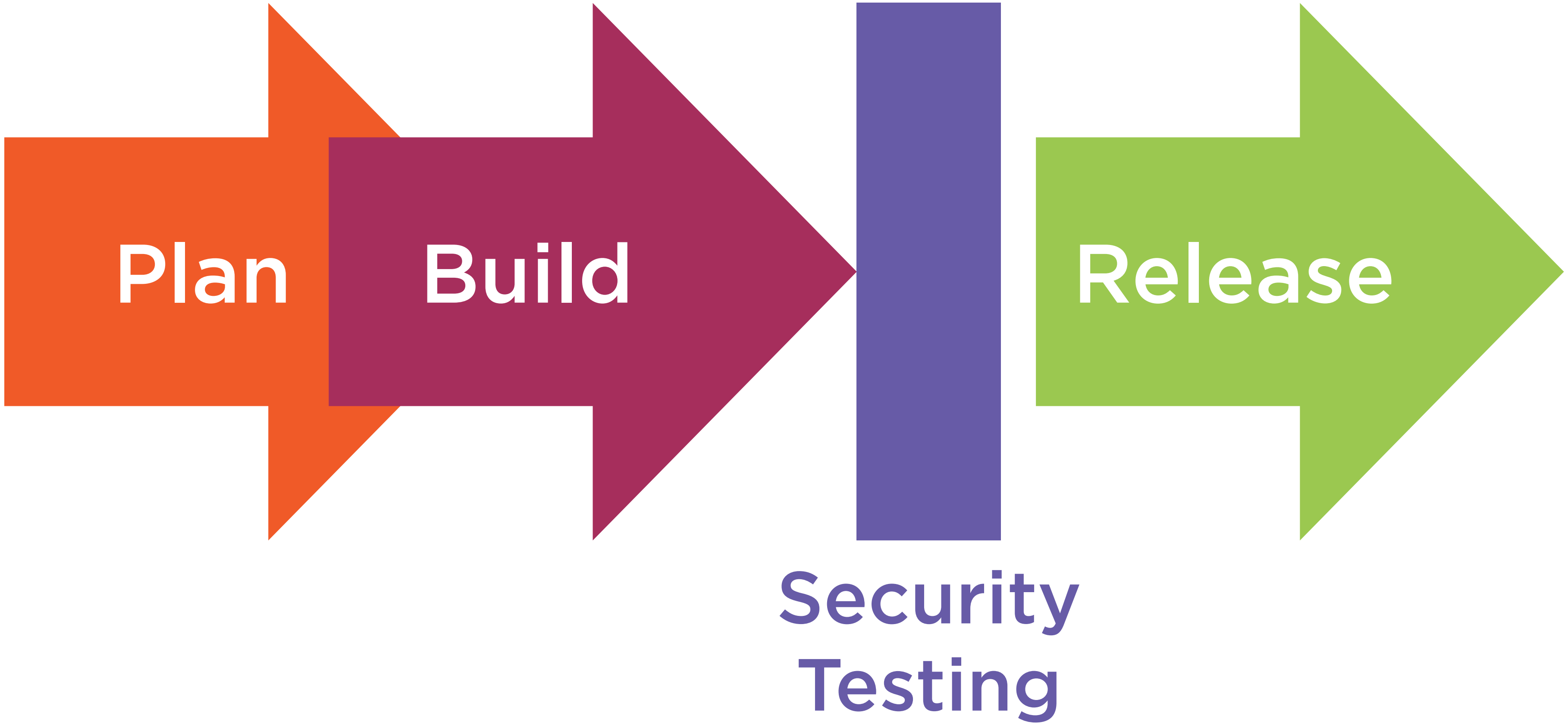
GitHub Marketplace

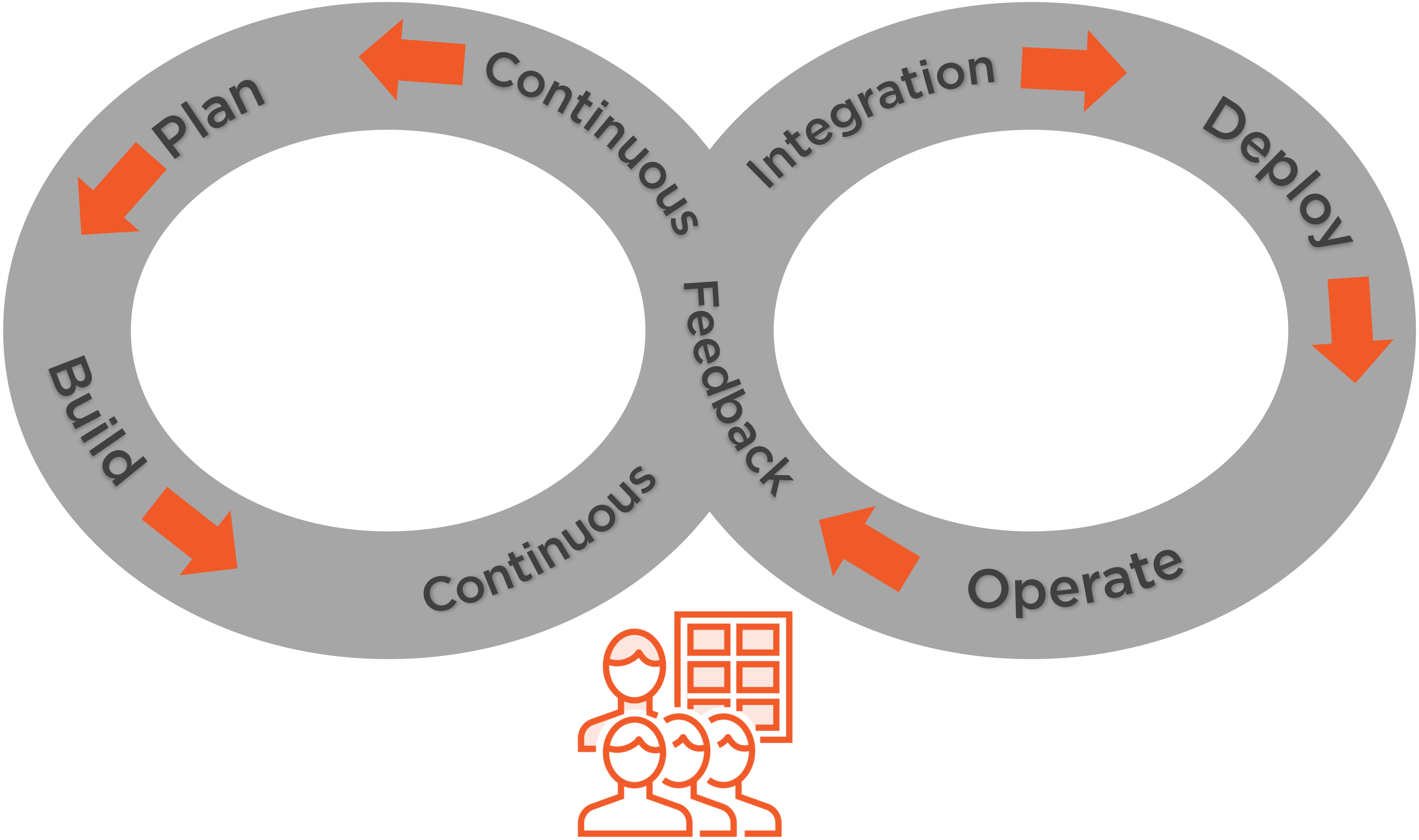
Runner

GitHub

Actions

DevSecOps and Shifting Security Left


Plan
Build
Release
Security
Testing



Plan

Continuous

Integration

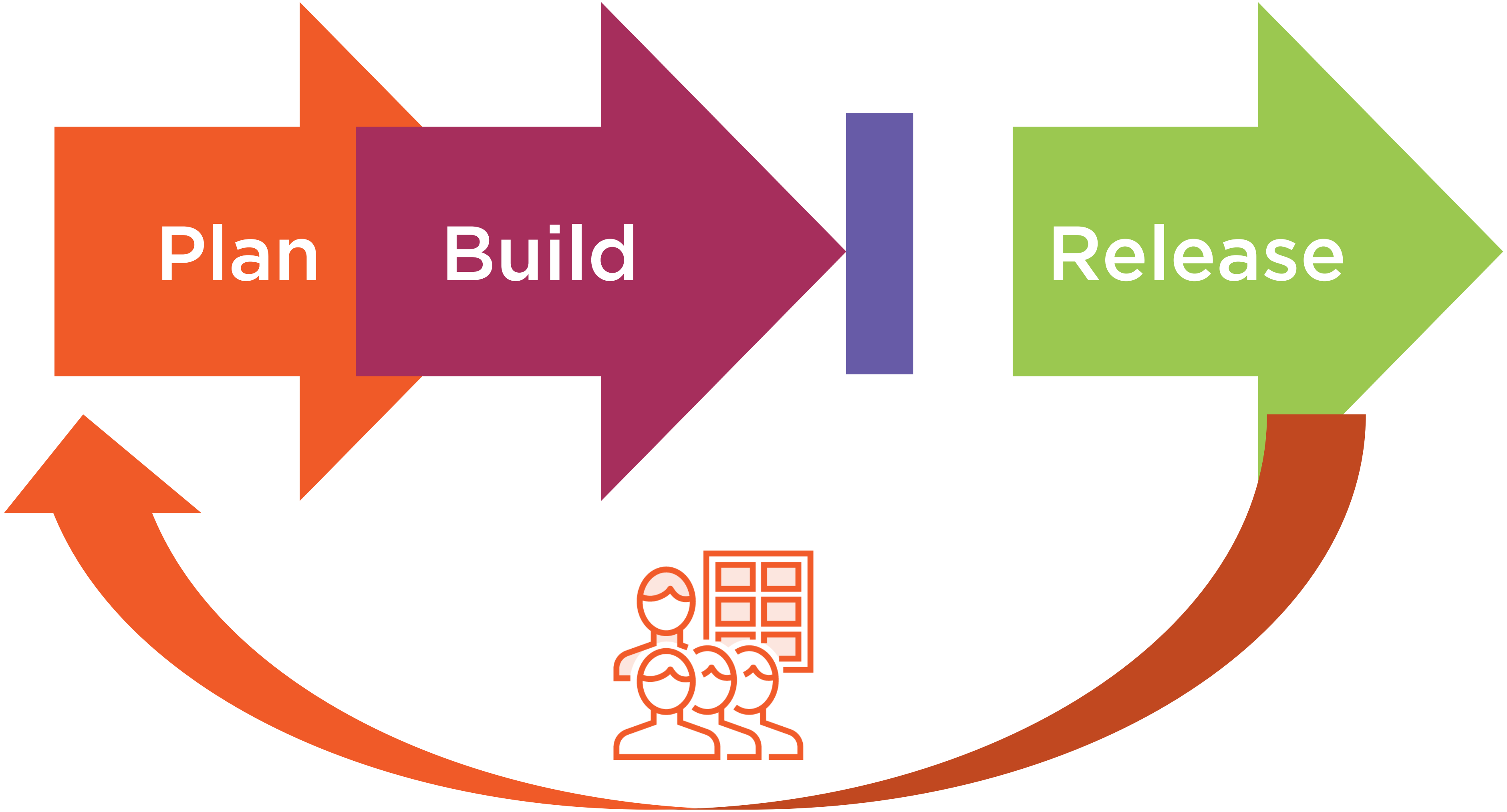
Deploy

Operate

Feedback

Continuous

Build


Plan
Build
Release

DevSecOps

Security as a priority

Security requirements

Configuration management

Automated testing

DevSecOps

Security as a priority

Security requirements

Configuration management

Automated testing

Static Application Security Testing (SAST)

Static Application Security Testing

- Examine code base for known patterns and vulnerabilities
- White Box testing
- During continuous integration (CI)
- During pull requests (PR)

GitHub Code Scanning

CodeQL queries used to identify patterns

- Semantic code analysis engine
- Object oriented programming language
- Use open-source queries
- Can write your own queries

GitHub Code Scanning

- Interoperable with other code scanning tools that output SARIF data
- Available for public repositories and private organization-owned repos
- Each run of code scanning consumes minutes of GitHub Actions

SonarQube

SonarQube

Freely available

Enterprise version available

Installed on a server

Executive-level reporting across projects

Support for plugins

SonarCloud

Software-as-a-Service

Centered on developers

Dynamic Application Security Testing (DAST)

Dynamic Application Security Testing

Testing a deployed application

Black Box testing

Identifies common security vulnerabilities

OWASP ZAP

Open Web Application Security Project

Zed Attack Proxy (ZAP)

Free and open source

OWASP ZAP Scan Types

Passive Scan

Examines the results returned

Spiders a site to discover pages

Doesn’t manipulate requests

Fast running

Good for Continuous Integration pipelines

Active Scan

Simulates techniques of hackers

Dynamic tests

Longer running

Good for nightly builds

OWASP ZAP

- Graphical user interface
- API and command line
- Requires Java runtime
- Windows, Linux or Mac
- Weekly Docker image

Hosting OWASP ZAP

Virtual Machine

Install ZAP exe

Call API from CI/CD workflow

Docker Container

Weekly download

Need to refresh container

Pre-built Task/Action

Azure DevOps Marketplace

GitHub Marketplace

Manual Integration of Zap into CI/CD

- Deploy Web Application
- Create / Pull OWASP Zap Docker Container
- Call Baseline or Full Scan
- Download Report File
- Transform Report Format
- Publish Report Results

Only perform penetration testing on apps you have **permission** to test!

Dependency Scanning for Security and Compliance

Libraries, packages, etc.

Dependencies you don’t maintain

Vulnerabilities are logged to internet databases

- National Vulnerability Database (NVD)

Dependency Scanning

Inventory of Dependencies in Project

Check Dependencies for Known Vulnerabilities

Check Versions of Dependencies in Use

Open-source Licenses in use by Dependencies

Open Source Initiative (OSI)

Most Popular / Widely-used Open Source Licenses

Apache License 2.0

BSD 3-Clause "New" or "Revised" license

BSD 2-Clause "Simplified" or "FreeBSD" license

GNU General Public License (GPL)

GNU Library or "Lesser" General Public License (LGPL)

MIT license

Mozilla Public License 2.0

Common Development and Distribution License

Eclipse Public License version 2.0

Permissive License

Guarantees the freedom to use, modify, and redistribute, and permits proprietary derivative works

MIT licenses

- .NET Core
- AngularJS
- JQuery

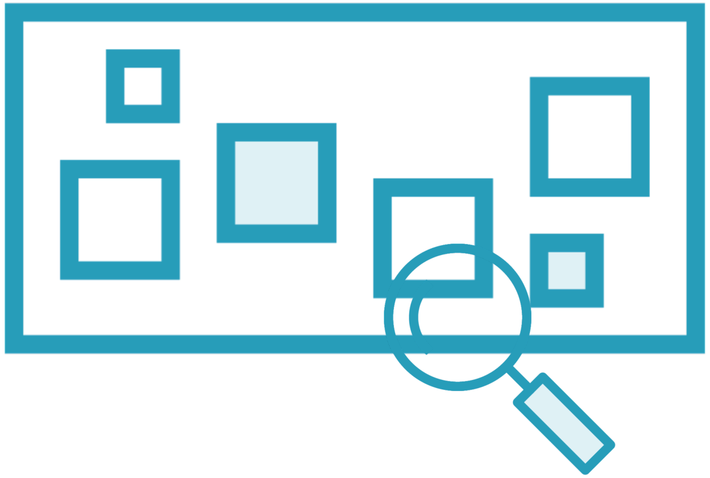
Copyleft License

Allow derivative works but require them to use the same license as the original work

e.g. GNU General Public License (GPL)

- Drupal
- Nant
- MariaDB
- MySQL
- Joomla
- Git
- Wordpress

Dependency Scanning of Containers



Container Scanning

Examines packages and dependencies in image layers

Scanner will often recommend a better base image when appropriate

Built-in scanning available in some container registries

Third-party offerings for scanning during CI/CD process

Container Registry Scanning Capabilities

Azure Container Registry

Uses Qualys scanning

Azure Security Center

Results can be queries through Graph API

Docker Enterprise

Scanning in Docker Trusted Registry

Docker Hub

Uses Snyk scanning

Repo scanning

TRIVY

Open-source container scanner

Aqua Security

Scan images in a container repository

Output scan results as SARIF

Scanning for Secrets

Credentials, API keys, connection strings

Should be stored outside code

Static code analysis

- Works on current code and history
- Matches regular expressions

Third-party scanning tools

GitHub Secret Scanning

Responding to Events with Git Hooks

Git Hooks

Run scripts when an event occurs

Client-side events or server-side events

- pre-commit
- post-commit
- pre-receive
- post-receive
- Other events available

.git/hooks

Up Next:
Automating Scans in Your CI/CD Pipeline